Army Research Laboratory

Benchmarking the Benchmarks

by Daniel M. Pressel and Jelani Clay

ARL-TR-2805 September 2002

NOTICES

Disclaimers

Army Research Laboratory
Aberdeen Proving Ground, MD 21005-5067

ARL-TR-2805 **September 2002**

Benchmarking the Benchmarks

Daniel M. Pressel
Computational and Information Sciences Directorate, ARL

Jelani Clay
Prairie View A & M University

Abstract

Benchmarks can be useful in estimating the performance of a computer system when it is not possible or practical to test out the new system with an actual workload. In the field of high performance computing, some common benchmarks are the various versions of Linpack, the various versions of the Numerical Aerospace Simulation Systems Division of NASA Ames Research Center (NAS) benchmarks, and the STREAMS benchmark, as well as older and less frequently referenced benchmarks such as the Livermore Loops. There are also those who recommend estimating the performance based solely on the peak speed of the computer systems. Unfortunately, the per processor levels of performance measured using these benchmarks can vary by 1 to 2 orders of magnitude for the same system. Therefore, one has to ask, which benchmark(s) should we be looking at? This report attempts to answer that question by comparing the measured performance for a variety of real world codes to the measured performance of the standard benchmarks when run of systems of interest to the Department of Defense (DOD) High Performance Computing Modernization Program.

Acknowledgments

This work was made possible through a grant of computer time from the Department of Defense (DOD) High Performance Computing Modernization Program (HPCMP).

The author would like to thank the following for their assistance in this effort:

- Dr. N. Radhakrishnan, Dr. Andrew Mark, Emma Grove, and the entire U.S. Army Research Laboratory (ARL)-Major Shared Resource Center (MSRC) Programming Environmental and Training (PET) office for their involvement with the intern program,
- Jerry Clarke and Dixie Hisley, of ARL, for their assistance mentoring Mr. Clay,
- Marek Behr, Rice University; Shirley Moore, University of Tennessee; Sirpa Saarinen, National Center for Supercomputing Application at the University of Illinois/Urbana-Champaign; A. Snavely, San Diego Supercomputer Center, University of California at San Diego; J. M. Levesque, IBM ACTC; P. Satya-narayana, Raytheon Systems Co.; and Steve Schraml and Csaba Zoltani (ARL) for their helpful communications,
- Kirk Kern of Silicon Graphics Inc. (SGI) for supplying benchmark results for the Origin 2000 and Origin 3000, and
- Susan Sassaman of Business Plus Corp. (BPC) for editorial services on this report.

INTENTIONALLY LEFT BLANK.

Contents

INTENTIONALLY LEFT BLANK.

List of Figures

INTENTIONALLY LEFT BLANK.

List of Tables

INTENTIONALLY LEFT BLANK.

1. Introduction

During the summer of the year 2000, as part of his student internship at the **ARL-MSRC**,* Jelani Clay, under the supervision of Daniel M. Pressel, investigated the following question: Which, if any, of the industry standard benchmarks adequately predict the performance of real world codes on systems of interest to the **DOD HPCMP**? Several benchmarks have been proposed for this purpose, including the following:

- the theoretical peak performance of the system,
- the current SPEC benchmarks,
- one or more of the Linpack family of benchmarks,
- the Livermore Loops,
- the STREAMS benchmark, and
- some of the **NAS** family of benchmarks.

We concluded that the SPEC benchmarks were primarily single-processor benchmarks aimed at workstation class systems and therefore deleted them from our list. Micro benchmarks that seemed to be aimed at measuring the performance of a specific feature of the architecture were deleted. This included benchmarks for **FFT**s, Matrix Multiply, various cache benchmarks, etc. It was also felt that the Livermore Loops were generally considered to be obsolete and rarely reported anymore. The final selection included the following benchmarks and datasets:

- the theoretical peak performance of the system,
- the Linpack Benchmark-Parallel when the data was available, supplemented with results for the Linpack N=1000 benchmark,
- the STREAMS benchmark, and
- the NAS **NPB** 2 benchmarks for the class B data set (BT, CG, LU, and SP), supplemented with results for the class A data set.

Following this, a search of conference papers and websites related to high performance computing was undertaken with the goal of finding published performance results for as wide a range of programs as possible. Unfortunately, this required us to be able to determine as precisely as possible the following three things:

---

* Definitions for boldface text can be found in the Glossary.

(1) What system was being used (e.g., simply knowing that the system was an SGI Origin 2000 with a R10000 processor or an IBM SP with a P2SC processor was not sufficient if we did not know the processor speed)?

(2) How many processors were used?

(3) What was the performance in **MFLOPS** per processor or some other unit that could readily be converted to this unit?

The problem was that many other excellent papers were missing one or more of these numbers. In rare instances, sufficient information existed from other sources that we were able to fill in the blanks. However, in an unfortunately large number of cases, we had to discontinue our search and proceed with our research.

After analyzing all of the data that was collected, we arrived at the following conclusions:

(1) The peak speed of the system is a particularly bad predictor of system performance.

(2) The Linpack benchmarks closely track the peak system speed and therefore suffer from the same failing.

(3) The STREAMS benchmark is primarily a serial benchmark and says very little about the scalability of the system. It also tends to underpredict the performance of single-processor runs.

(4) The NAS benchmarks support several data sets (classes A—small, B—medium, C—large, and W—"workstation") and come in four main flavors (NPB 1—pencil and paper, NPB 2—MPI, and experimental versions based on HPF and OpenMP). The NPB 2 results produce a range of performance numbers which seem to correspond closely with the performance results seen by many real world codes.

2. Methodology

The ideal methodology is to determine which systems are located at the major sites of interest (e.g., systems located at the MSRCs and the larger **DCs**) to the target audience (e.g., the Users Group for the DOD HPCMP). Next, one must try to determine which benchmarks are the most relevant to the problem domain in question. In the case of this report, the problem domain is **HPC** applications—particularly those applications that are routinely run using at least 100 processors for a single job. As such, we investigated a large number of commonly referenced benchmarks and found:

- The TPC benchmarks are heavily oriented towards database and not HPC applications and are therefore not relevant to this study.
- The SPEC benchmarks are relatively small serial benchmarks aimed at the desktop/deskside market and, again, lacked relevancy.
- Benchmarks such as Dhrystone and Whetstone are obsolete and rarely mentioned anymore. Furthermore, they were designed to measure the total instruction execution rate, not just the floating point execution rate, on single processor departmental servers circa 1980s.
- Benchmarks such as the four "FLOPS" benchmarks maintained by Alfred Aburto of the Naval Ocean Systems Center, San Diego, CA, are slightly better in that they only deal with floating point operations. However, they still fail to address the need for a parallel benchmark for HPC applications.
- Similarly, we felt that benchmarks based on narrowly defined computational kernels (e.g., matrix multiply or FFTs) were too narrow in scope to be used to benchmark an entire machine.
- Micro benchmarks (e.g., those designed to investigate the caches) can be quite useful, but not for this study.
- Livermore Loops looked more promising, but they were found to be dated and rarely referenced in recent literature.

Therefore, we settled on the following set of benchmarks:

- the theoretical peak performance of the system,
- the Linpack Benchmark-Parallel when the data was available, supplemented with results for the Linpack N=1000 benchmark,
- the STREAMS benchmark, and
- the NAS **NPB** 2 benchmarks for the class B data set (BT, CG, LU, and SP), supplemented with results for the class A data set.

We then proceeded to collect the necessary data. Where data are missing, one might consider personally performing the runs. We chose not to take this approach and instead have attempted to estimate the missing data points using the following approaches:

- When Linpack-Parallel results were not readily available, we attempted to use Linpack N=1000 results. If neither were available, but results from a similar system from the same vendor (e.g., IBM P2SC 120 MHz is similar to the IBM P2SC 135 MHz) were available, then the results from the similar system were used, with the performance scaled based on the clock rates.
- When NAS NPB 2 results for the class B data set were not available, results for the class A data set were used.

- Once the NPB 2 data set was selected, if results for a run using the correct number of processors could not be found, then results for the closest number of processors reported were used. In some cases, this was 1. This could have potentially presented a serious problem when comparing this result to runs involving out to 100 or more processors. Fortunately, in the case of the SUN HPC 10000, we were able to substitute results for the OpenMP version of this benchmark. Hopefully, this will make for more realistic comparisons.
- Again, it was sometimes necessary to extrapolate results from measured systems to similar systems where the data was missing. The most questionable use of this approach involved the four IBM SP systems with Power 3 processors. Fortunately, as these systems have matured, additional benchmark results have become available.
- For the STREAM benchmark, it was generally possible to obtain single processor runs. When this was not the case, and keeping in mind that this benchmark was designed to primarily measure the performance of the memory system and not the processor, we used results for a similar system without any scaling. Even so, in the case of the IBM SP with Power 3 processors, this may not have been very accurate due to the significant differences in architecture of the memory systems for the different types of nodes. Another issue was that for any **SMP** or system with SMP nodes, running a job on a single processor with the other processors in the system/node idle would overstate the available memory bandwidth on a per-processor basis and therefore skew the results to some extent.

Once we had the benchmark numbers, those that were not already in MFLOPS/processor terms were converted to that format. For the NAS benchmarks, we attempted to collect the results for two ranges of processor counts—100-200 processors and more than 200 processors. Some systems either didn't go that large or had not been benchmarked for the larger configurations. In those cases, we had to extrapolate the data as was previously mentioned.

The results for the real world codes were collected from a variety of sources, including conference proceedings and runs done by employees of ARL. These numbers were then grouped into three groups, depending on the processor counts—1-99 processors, 100-200 processors, and more than 200 processors. Again, the results were expressed in terms of MFLOPS/processor. No attempt was made to extrapolate results to systems/system configurations where data was missing. In many cases, it was clear that the researchers had not continued to higher processor counts either because they had run out of processors and/or because their jobs were no longer scaling well. In either case, extrapolating the results did not seem to be worthwhile.

3. Observations and Results

Figures 1 and 2 and Table 1 compare the benchmark data with the peak speed of the processors. The Linpack results closely track the peak system speed, although they have the added benefit of tracking the scalability of the system for certain classes of codes. Even so, they tend to overpredict the performance in a similar fashion to using the peak speed. In general, the NAS and STREAM benchmark results were significantly slower than the Linpack benchmark results.*

When comparing the NAS and STREAM benchmark results, it was not clear how much of a difference there was between the results for these two sets of benchmarks. Therefore, we constructed Figure 3 and Table 2 to compare the single processor performance of the NAS benchmarks to the results for the STREAM benchmarks. One complication in compiling this data is that due to memory constraints, most vendors did not report single processor runs for the NAS benchmarks. Therefore, we had to use the runs done with the smallest number of processors, in the 1–16 processor range. From this, the following two things became clear:

(1) The single processor performance for the NAS benchmarks was, in general, significantly greater than what the STREAM benchmark was predicting.

(2) By comparing the data from Table 1 (Figures 1 and 2) with the data from Table 2 (Figure 3) for the NAS benchmarks, one can clearly see the importance of taking the system interconnect into consideration. One problem with this was that each code would interact with the system interconnect in its own way, making it difficult to offer sweeping generalizations. For this reason, we decided not to pursue the STREAM benchmark further. Additionally, the importance of separating out the benchmark runs and real world runs into groups based on the number of processors being used became all too clear.†

* The NAS benchmarks support several data sets (classes A—small, B—medium, C—large, and W—"workstation") and come in four main flavors (NPB 1—pencil and paper, NPB 2—MPI, and experimental versions based on HPF and OpenMP). We found that the NPB 1 results were usually significantly faster than the NPB 2 results and probably should be considered to be overly optimistic for most real world codes. Results for HPF and OpenMP were not generally available for most systems and therefore were not analyzed. The NPB 2 results produce a range of performance numbers that seem to correspond closely with the performance results seen by many real world codes. The main drawback to using the NPB 2 results is the difficulty of obtaining numbers for new systems, since the NAS group at NASA Ames has not recently posted new results to their website.

† If the reader compares the relative values for the NAS CG and the STREAM benchmark results, one will see that the CG benchmark performs much better when using only a few processors (on a per processor basis), while the STREAM benchmark is virtually unaffected by the number of processors used. Therefore, when looking for a reasonable lower bound on the performance of parallel jobs, the NAS CG benchmark looks like it will be a better choice.

Figures 4–7 and Table 3 contain our results from mining the web and a variety of conference proceedings for results involving real world codes. One can easily see that for many of the systems a wide range of performance was reported (e.g., one order of magnitude). To simplify the comparison, the benchmark results and the results for real world codes were expressed in terms of ranges of performance, with these numbers appearing in Figures 7–9 and Table 4. This allowed us to clearly see that in many cases, the Linpack results significantly overstated the performance that one was likely to achieve with real world codes on modern HPC systems. Even so, a small number of extremely well-tuned codes exhibited levels of performance that were comparable to those reported for the Linpack benchmark. In most cases, the results for the NAS benchmarks as a group were a better predictor. Unfortunately, without a more specific knowledge of the algorithms involved in the real world codes, it was difficult to be more precise as to what level of performance any single code would exhibit. Even then, the results clearly indicated that differences between two data sets of fixed size could affect the scalability and performance of the same code on the same system. There was also the additional complication of how much time, effort, and skill the author of a real world code could contribute when writing or porting a program.

4. Conclusions

When looking at the NAS NPB 2 benchmarks (BT, CG, LU, and SP) as a group, their range of performance on a particular system of a particular size range seems to be a good predictor of performance by well-tuned real world codes on the same system. In most cases, this metric will be a better choice than using either the STREAM or the Linpack benchmarks. We believe that the class B data set for the NPB 2 benchmarks is, in general, the best choice; although for smaller system sizes, class A may also be appropriate. Similarly, for larger system sizes, the rarely reported class C data set may be a better choice.

There were two major problems in carrying out this study:

(1) People have stopped reporting the NAS benchmarks and in some cases, the STREAM and/or Linpack benchmarks, for new systems. We recommend that efforts be made to measure and publicly disseminate the performance numbers for these benchmarks for as wide a range of systems/system configurations as is practical.

(2) Even when the author of a paper is primarily interested in the science aspect and not the performance when measured in MFLOPS, it would still be helpful to have such numbers reported.

It is also important to note that this study has some important limitations. Topping the list is the question of input/output. We feel that input/output is a sufficiently complicated issue that is best left to another study. The same holds true for issues such as usability and system stability. The results for the **MIMD** version of the F3D code demonstrate that if one attempts to implement a very fine grained level of parallelism using MPI and an **MPP** with a moderate-to-large message latency, the performance will suffer to the point that none of the benchmarks will accurately predict the level of performance. It is best if one can avoid fine grained levels of parallelism whenever possible. When that is not possible, the use of OpenMP on a shared memory platform or a low-latency message-passing library such as SHMEM on an MPP with a relatively low-message latency are better choices.

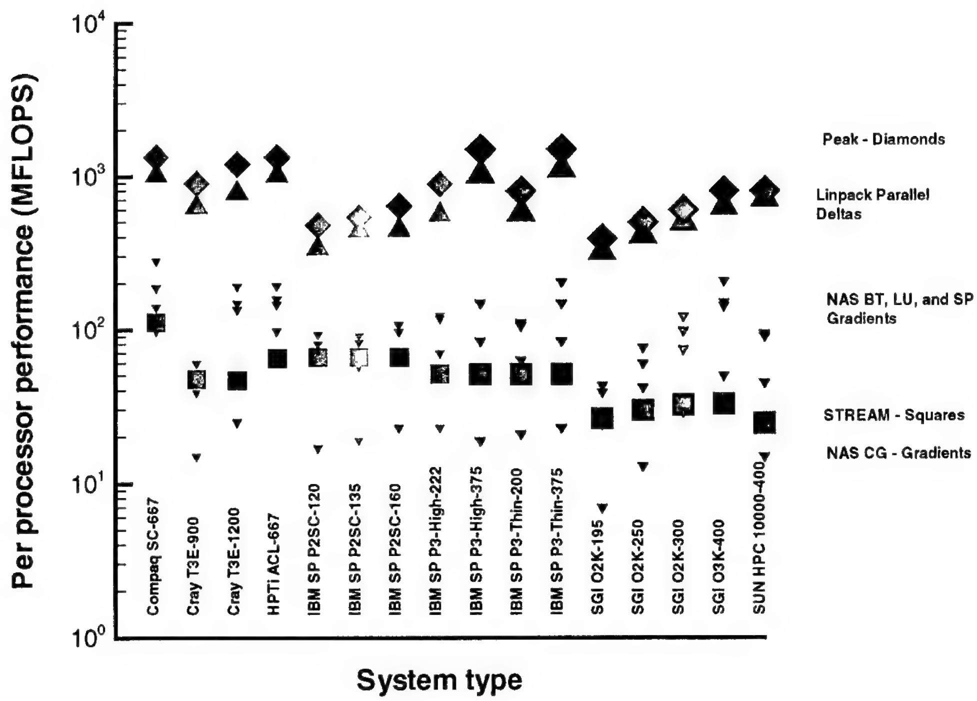

Figure 1. Comparison of commonly used HPC benchmarks (100–200 processors).

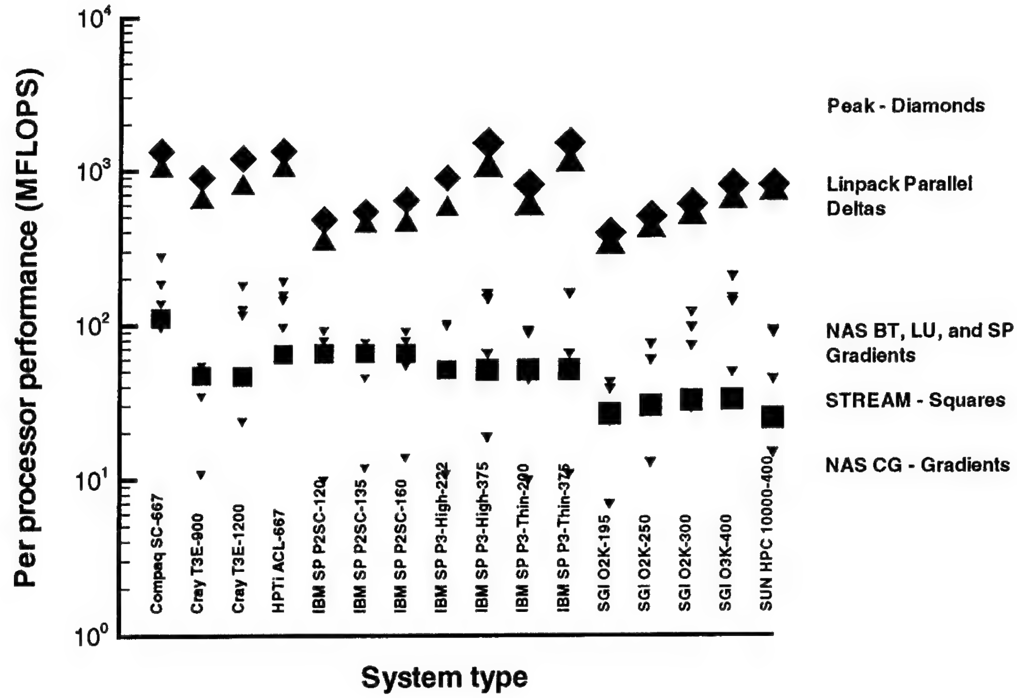

Figure 2. Comparison of commonly used HPC benchmarks (>200 processors).

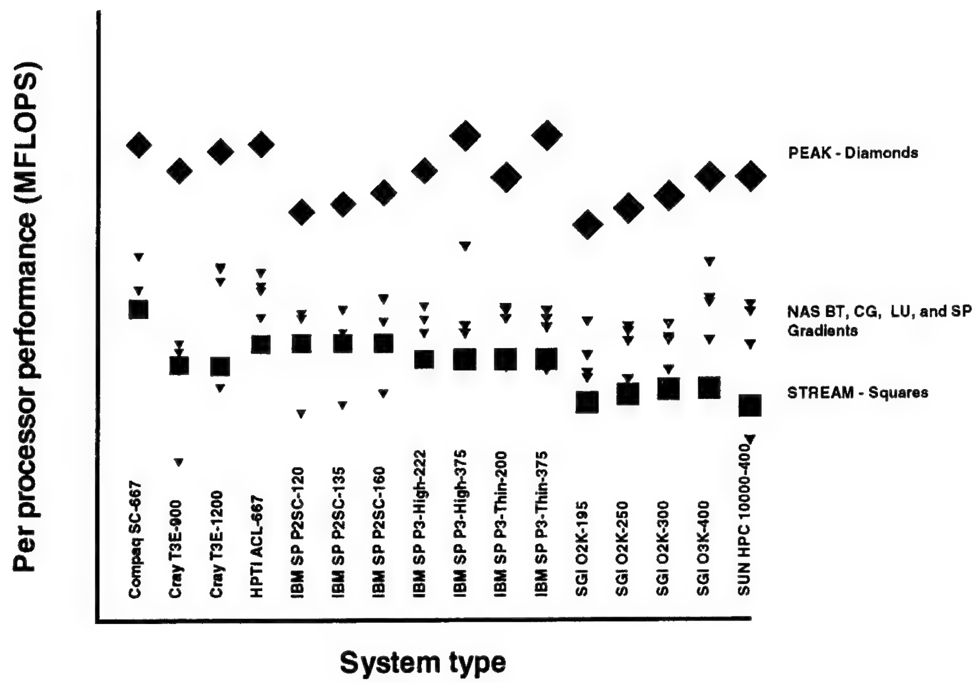

Figure 3. Comparison of commonly used HPC benchmarks (1–16 processors).

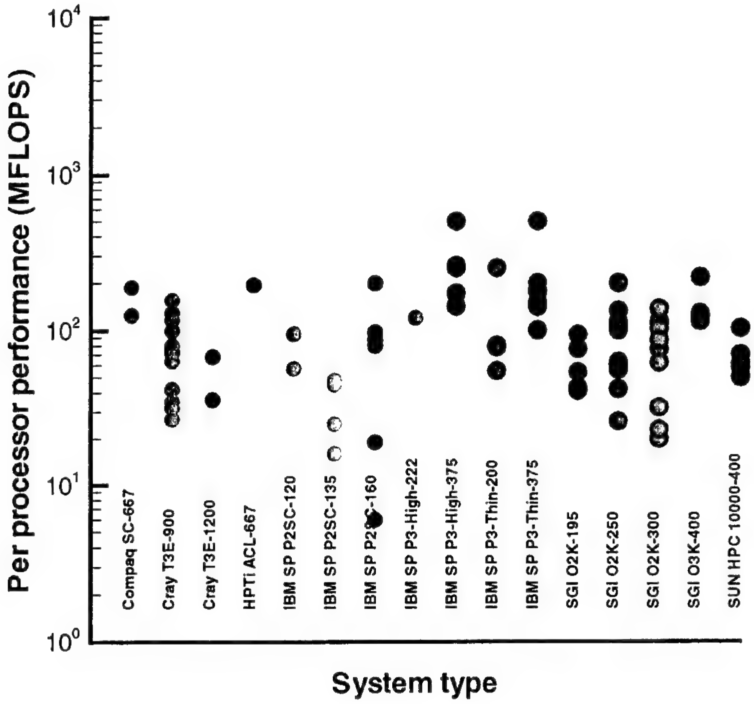

Figure 4. Performance results for a wide range of real world codes (<100 processors).

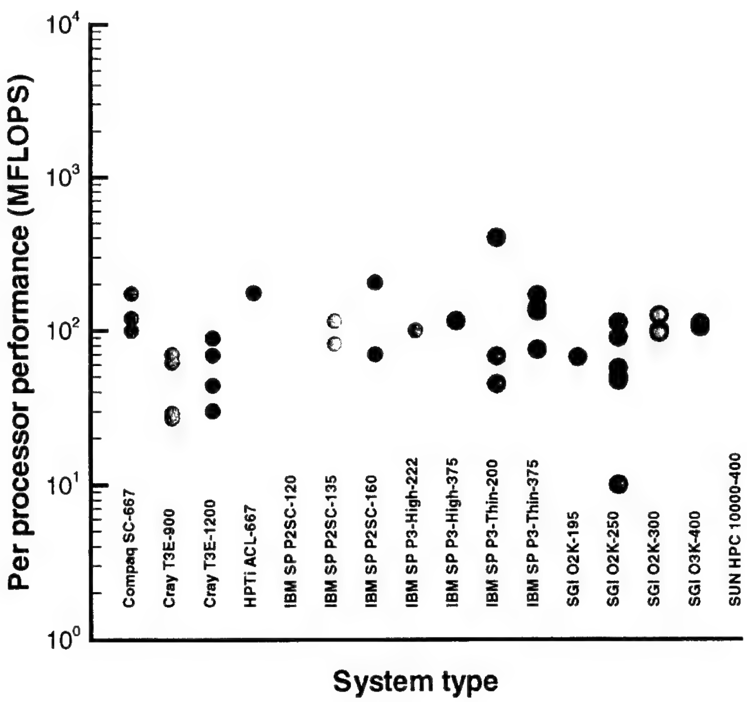

Figure 5. Performance results for a wide range of real world codes (100–200 processors).

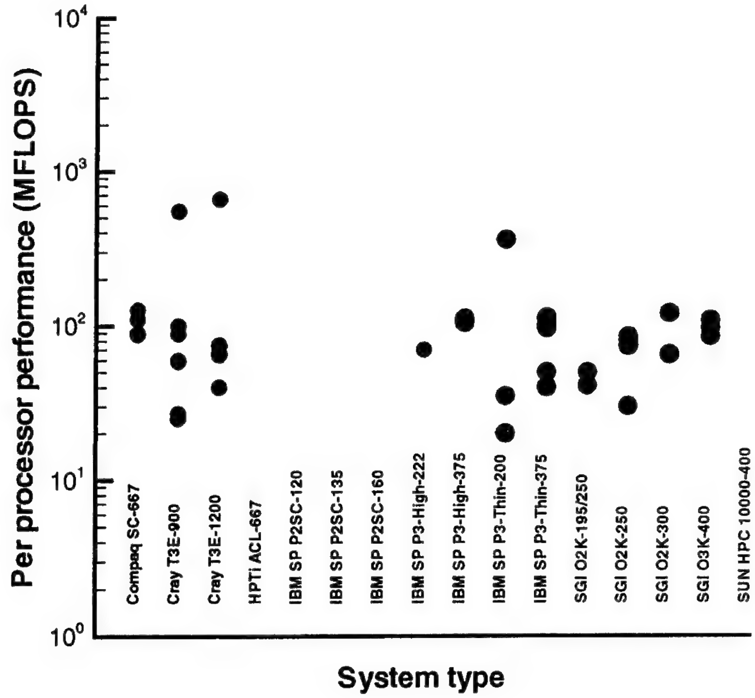

Figure 6. Performance results for a wide range of real world codes (>200 processors).

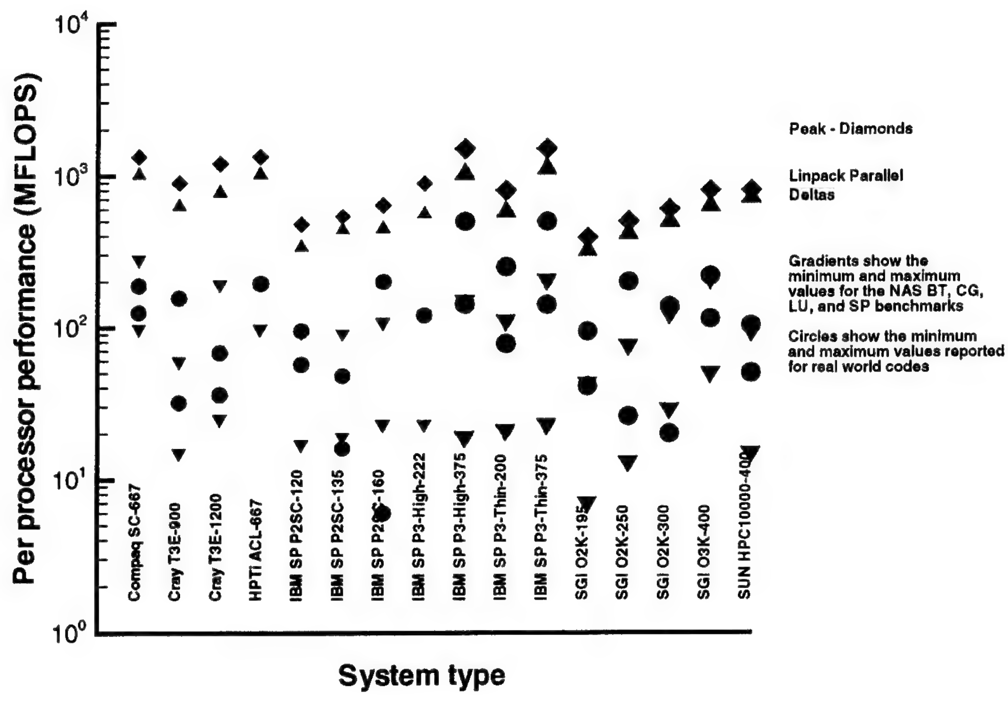

Figure 7. Comparison of commonly used HPC benchmarks to real world codes (<100 processors).

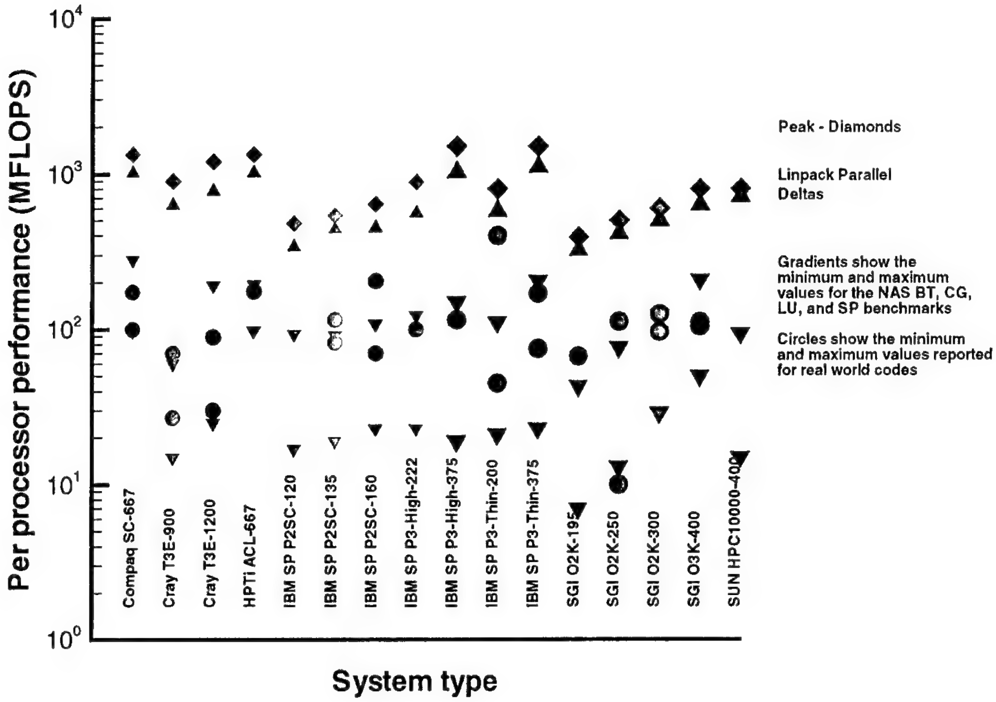

Figure 8. Comparison of commonly used HPC benchmarks to real world codes (100–200 processors).

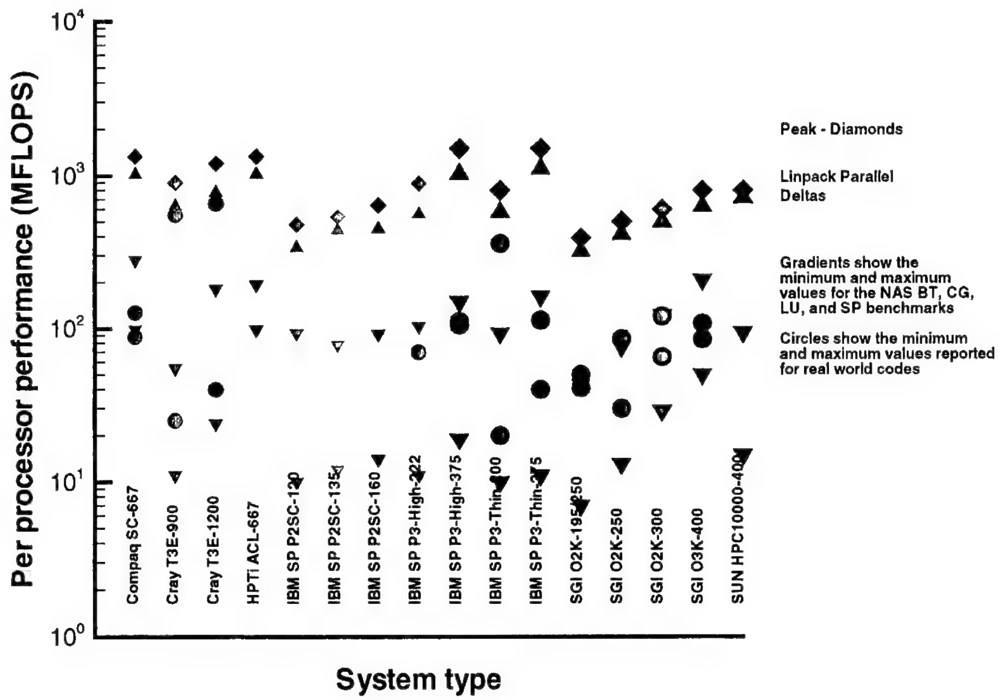

Figure 9. Comparison of commonly used HPC benchmarks to real world codes (>200 processors).

Table 1. The performance of commonly used systems within the DOD HPCMP on commonly referenced benchmarks.

System Type	Stream Triad 1 Processor		Linpack Parallel per Processor		NAS Class B per Processor (MFLOPS)										Peak per Processor
					100–200 Processors					>200 Processors					
	(MFLOPS)	Reference	(MFLOPS)	Reference	BT	CG	LU	SP	Reference	BT	CG	LU	SP	Reference	(MFLOPS)
Compaq SC-667	111.5	[1]	1015	[2]	188	98	281	140	[5], [37], est.	188	98	281	140	[5], [37] est.	1334
Cray T3E-900	47.3	[1]	632	[2]	51	15	60	39	[4]	50	11	55	35	[4]	900
Cray T3E-1200	46.5	[1]	776	[2]	66	12	72	49	[4], est.	66	12	72	49	[4], est.	1200
HPTi ACL-667	64.8	[1], est.	1015	[2], est.	194	98	158	147	[5], est.	194	98	158	147	[5], est.	1334
IBM SP P2SC-120	65.6	[1]	338	[2]	93	17	80	62	[4], est.	93	10	80	62	[4], est.	480
IBM SP P2SC-135	65.6	[1], est.	440	[3]	91	19	82	57	[4], est.	78	12	68	46	[4], est.	540
IBM SP P2SC-160	65.6	[1], est.	447	[2]	108	23	97	68	[4]	92	14	80	55	[4]	640
IBM SP P3-HIGH-222	51.2	[1]	560	[2]	118	23	123	70	[6], est.	100	11	103	50	[6], est.	888
IBM SP P3-HIGH-375	51.2	[1], est.	1023	[2]	149	19	150	84	[6], [36] est.	161	19	150	66	[6], [36] est.	1500
IBM SP P3-THIN-200	51.2	[1], est.	576	[2]	106	21	111	63	[6]	90	10	93	45	[6]	800
IBM SP P3-THIN-375	51.2	[1], est.	1106	[2]	149	23	205	84	[6]	161	11	162	66	[6]	1500
SGI O2K-195	26.4	[1]	322	[2]	43	7	39	24	[7], est.	43	7	39	24	[7], est.	390
SGI O2K-250	29.8	[1]	412	[2]	76	13	60	42	[8]	76	13	60	42	[8], est.	500
SGI O2K-300	32.3	[1]	498	[2]	122	29	98	74	[38], [38], est.	122	29	98	74	[38], est.	600
SGI O3K-400	32.8	[1]	683	[38]	143	50	208	151	[38], est.	143	50	208	151	[38], est.	800
SUN HPC10000-400	24.7	[1]	713	[2]	94	15	90	45	[35], est.	94	15	90	45	[35], est.	800

Table 2. The serial performance of commonly used systems within the DOD HPCMP on commonly referenced benchmarks.

System Type	System Triad 1 Processor		NAS Class B per Processor (MFLOPS) 1–16 Processors					Peak per Processor
	(MFLOPS)	Reference	BT	CG	LU	SP	Reference	(MFLOPS)
Compaq SC-667	111.5	[1]	150	120	250	150	[5], [37], est.	1334
Cray T3E-900	47.3	[1]	58	11	66	44	[4]	900
Cray T3E-1200	46.5	[1]	67	10	79	50	[4]	1200
HPTi ACL-667	64.8	[1], est.	194	98	158	147	[5], est.	1334
IBM SP P2SC-120	65.6	[1]	104	23	97	72	[4], est.	480
IBM SP P2SC-135	65.6	[1], est.	111	26	109	78	[4], est.	540
IBM SP P2SC-160	65.6	[1], est.	131	31	129	92	[4]	640
IBM SP P3-HIGH-222	51.2	[1]	116	77	78	95	[11], [12], est.	888
IBM SP P3-HIGH-375	51.2	[1], est.	77	56	288	86	[6], [36], est.	1500
IBM SP P3-THIN-200	51.2	[1], est.	108	44	96	84	[6]	800
IBM SP P3-THIN-375	51.2	[1], est.	77	45	224	86	[6]	1500
SGI O2K-195	26.4	[1]	55	39	92	42	[4]	390
SGI O2K-250	29.8	[1]	79	38	85	68	[7], [8]	500
SGI O2K-300	32.3	[1]	72	44	88	69	[7], [9]	600
SGI O3K-400	32.8	[1]	130	69	224	122	[38]	800
SUN HPC10000-400	24.7	[1]	118	15	106	64	[35], est.	800

Table 3. The performance of commonly used systems within the DOD HPCMP as reported for real world codes.

System Type	Program Name	CTA	Number of Processors Used	Performance per Processor (MFLOPS)	Reference
Jobs using less than 100 processors					
Compaq SC-667	CCM/MP-2D	CWO	64	125	[39]
	MM5	CWO	64	188	[31]
Cray T3E-900	Paratec	CCM	64	117	[16]
	Paratec	CCM	64	156	[16]
	Ocean/Wallcraft	CWO	96	32	[17]
	NAMD	CCM	64	64	[19]
	CCM/MP-2D	CWO	64	35	[20]
	CCM/MP-2D	CWO	64	27	[20]
	Ocean/Wallcraft	CWO	60	75	[24]
	PCM	CWO	64	32	[25]
	CCM3	CWO	64	42	[25]
	FE-MIMD	CFD	49	130	[27]
	Uncle	CFD	50	72	[30]
	PSTSWM	CWO	1	80–100	[40]
	SUBOFF	CFD	50	72	[50]
Cray T3E-1200	RIEMANN		64	68	[14]
	F3D-MIMD	CFD	88	36	[28]
HPTI ACL-667	MM5	CWO	64	195	[31]
IBM SP P2SC-120	CG+Schwarz/Rich.	CFD	64	57	[16]
	FUN3D	CFD	80	95	[45]
IBM SP P2SC-135	Overflow	CFD	24	16	[23]
	Overflow	CFD	55	45	[23]
	Overflow	CFD	61	25	[23]
	MM5	CWO	64	48	[49]

Table 3. The performance of commonly used systems within the DOD HPCMP as reported for real world codes (continued).

System Type	Program Name	CTA	Number of Processors Used	Performance per Processor (MFLOPS)	Reference
Jobs using less than 100 processors					
IBM SP P2SC-160	Ocean/Wallcraft	CWO	60	80	[25]
	F3D-MIMD	CFD	88	6	[28]
	dDNS	CFD	?	200	[51]
	reservoir	CFD	16	87	[52]
	MM5	CWO	64	97	[53]
	cocoa	CFD	24	19	[54]
IBM SP P3-HIGH-222	Ocean/Wallcraft	CWO	60	120	[25]
IBM SP P3-HIGH-375	CTH	CSM	1	259	[33]
	CTH	CSM	32	172	[33]
	CTH	CSM	64	142	[33]
	PSTSWM	CWO	1	250–500	[40]
	PSTSWM	CWO	16	250–500	[40]
IBM SP P3-THIN-200	MM5	CWO	64	78	[32]
	CCM/MP-2D	CWO	64	55	[39]
	PSTSWM	CWO	1–2	80–250	[40]
IBM SP P3-THIN-375	Ocean/Wallcraft	CWO	60	180	[25]
	MM5	CWO	64	141	[32]
	CTH	CSM	64	150	[33]
	CCM/MP-2D	CWO	64	100	[39]
	PSTSWM	CWO	1	200–500	[40]
	PSTSWM	CWO	4	175–500	[40]
SGI O2000-195	CG+Schwarz/Rich.	CFD	64	94	[16]
	Ocean/Wallcraft	CWO	16	43	[18]
	F3D-SMP	CFD	88	54	[28]
	F3D-SMP	CFD	88	76	[28]
	CFDSHIP-IOWA	CFD	52	41	[30]

Table 3. The performance of commonly used systems within the DOD HPCMP as reported for real world codes (continued).

System Type	Program Name	CTA	Number of Processors Used	Performance per Processor (MFLOPS)	Reference
Jobs using less than 100 processors					
SGI O2000-250	DFT	CCM	64	100	[13]
	DFT	CCM	80	26	[13]
	DFT	CCM	75	63	[13]
	ZEUS	CFD	96	61	[15]
	CG+Schwarz/Rich.	CFD	64	106	[16]
	CG+Schwarz/Rich.	CFD	64	133	[16]
	NAMD	CFD	80	101	[20]
	CCM/MP-2D	CWO	64	63	[21]
	CCM/MP-2D	CWO	64	56	[21]
	PCM	CWO	64	42	[26]
	CCM3	CWO	64	60	[26]
	PSTSWM	CWO	1	100–200	[40]
	PSTSWM	CWO	64	100–200	[40]
	quark	?	64	113	[55]
SGI O2000-300	Ocean/Wallcraft	CWO	60	110	[25]
	F3D-SMP	CFD	88	76	[27]
	F3D-SMP	CFD	88	113	[27]
	F3D-MIMD	CFD	88	20	[28]
	MM5	CWO	64	137	[32]
	CTH	CSM	96	62	[33]
	Unstructured	CFD	64	23–32	[41]
	PAM-CRASH	CSM	32	102	[46]
	PAM-CRASH	CSM	64	86	[46]
SGI O3000-400	CTH	CSM	64	114	[32]
	PAM-CRASH	CSM	96	128	[46]
	MM5	CWO	64	218	[31]
	F3D-SMP	CFD	88	117	
	F3D-SMP	CFD	88	122	
SUN E10000-400	Ocean/Wallcraft	CWO	60	70	[25]
	F3D-SMP	CFD	64	58	[27]
	F3D-SMP	CFD	64	103	[27]
	CTH	CSM	64	61	[33]
2 * SUN E10000-400	CTH	CSM	96	50	[33]

Table 3. The performance of commonly used systems within the DOD HPCMP as reported for real world codes (continued).

System Type	Program Name	CTA	Number of Processors Used	Performance per Processor (MFLOPS)	Reference
Jobs using 100–200 processors					
Compaq SC-667	CCM/MP-2D	CWO	128	100	[39]
	CCM/MP-2D	CWO	128	120	[39]
	MM5	CWO	128	174	[31]
Cray T3E-900	NAMD	CCM	128	62	[20]
	CCM/MP-2D	CWO	128	29	[21]
	CCM/MP-2D	CWO	128	27	[21]
	Ocean/Wallcraft	CWO	110	70	[25]
Cray T3E-1200	RIEMANN		128	69	[14]
	F3D-MIMD	CFD	128	30	[28]
	FE-MIMD	CFD	128	89	[28]
	F3D-MIMD	CFD	128	44	[28]
HPTI ACL-667	MM5	CWO	128	176	[31]
IBM SP P2SC-135	Maxwell	CEM	128	82–115	[44]
IBM SP P2SC-160	Ocean/Wallcraft	CWO	110	70	[25]
	Lightning	CEM	125	205	[42]
IBM SP P3-HIGH-222	Ocean/Wallcraft	CWO	110	100	[25]
IBM SP P3-HIGH-375	CTH	CSM	128	115	[33]
IBM SP P3-THIN-200	MM5	CWO	128	68	[32]
	CCM/MP-2D	CWO	128	45	[39]
	WSSMP	GENERAL	128	400	[47]
IBM SP P3-THIN-375	Ocean/Wallcraft	CWO	110	170	[25]
	MM5	CWO	128	133	[32]
	CTH	CSM	128	140	[33]
	CCM/MP-2D	CWO	128	75	[39]
SGI O2000-195	F3D-SMP	CFD	120	67	[28]

Table 3. The performance of commonly used systems within the DOD HPCMP as reported for real world codes (continued).

System Type	Program Name	CTA	Number of Processors Used	Performance per Processor (MFLOPS)	Reference
Jobs using 100–200 processors					
SGI O2000-250	DFT	CCM	128	90	[13]
	DFT	CCM	135	10	[13]
	DFT	CCM	100	57	[13]
	ZEUS	CFD	192	47	[15]
	CCM/MP-2D	CWO	128	49	[21]
	CCM/MP-2D	CWO	128	50	[21]
	PPM	CFD	128	112	[43]
SGI O2000-300	Ocean/Wallcraft	CWO	110	100	[25]
	F3D-SMP	CFD	124	96	[27]
	MM5	CWO	120	125	[32]
SGI O3000-400	CTH	CSM	128	111	[34]
	F3D-SMP	CFD	128	104	
Jobs using more than 200 processors					
Compaq SC-667	CCM/MP-2D	CWO	256	90	[39]
	CCM/MP-2D	CWO	256	110	[39]
	MM5	CWO	256	127	[31]
	MM5	CWO	512	88	[31]
Cray T3E-900	NAMD	CCM	256	59	[20]
	CCM/MP-2D	CWO	256	25	[21]
	CCM/MP-2D	CWO	256	27	[21]
	Raleigh-Benard	CFD	1024	100	[22]
	Magnetism	CCM	512	552	[24]
	Ocean/Wallcraft	CWO	240	60	[25]
	Ocean/Wallcraft	CWO	1152	89	[29]
Cray T3E-1200	RIEMANN		256	67	[14]
	RIEMANN		512	65	[14]
	RIEMANN		1024	66	[14]
	RIEMANN		1490	66	[14]
	FUN3D	CFD	512	40	[19]
	FUN3D	CFD	1024	75	[19]
	Magnetism	CCM	1024	657	[24]

Table 3. The performance of commonly used systems within the DOD HPCMP as reported for real world codes (continued).

System Type	Program Name	CTA	Number of Processors Used	Performance per Processor (MFLOPS)	Reference
Jobs using more than 200 processors					
IBM SP P3-HIGH-222	Ocean/Wallcraft	CWO	240	70	[25]
IBM SP P3-HIGH-375	CTH	CSM	256	111	[33]
	CTH	CSM	480	109	[33]
	CTH	CSM	512	105	[33]
IBM SP P3-THIN-200	CCM/MP-2D	CWO	256	35	[39]
	CCM/MP-2D	CWO	512	20	[39]
	WSSMP	GENERAL	256	360	[48]
IBM SP P3-THIN-375	Ocean/Wallcraft	CWO	240	110	[25]
	MM5	CWO	256	96	[32]
	CTH	CSM	256	113	[33]
	CTH	CSM	512	101	[33]
	CCM/MP-2D	CWO	256	50	[39]
	CCM/MP-2D	CWO	512	40	[39]
SGI O2000-195/250	F3D-SMP	CFD	208	50	
	F3D-SMP	CFD	192	41	
SGI O2000-250	DFT	CCM	256	74	[13]
	ZEUS	CFD	256	30	[15]
	Overflow-MLP	CFD	256	80	[35]
	quark	?	250	85	[55]
SGI O2000-300	Ocean/Wallcraft	CWO	240	65	[25]
	Ocean/Wallcraft	CWO	512	65	[25]
	Overflow-MLP	CFD	512	120	[34]
SGI O3000-400	CTH	CSM	256	96	[32]
	F3D-SMP	CFD	232	85	
	F3D-SMP	CFD	248	108	

Table 4. A comparison of benchmark results to reported performance levels for real world codes for commonly used systems within the DOD HPCMP.

System Type	Linpack Parallel per Processor (MFLOPS)	NAS Class B per Processor (MFLOPS)		Peak per Processor (MFLOPS)	Per Processor Performance Ranges for Production Codes (MFLOPS)		
		100–200 Processors Performance Range	>200 Processors Performance Range		<100 Processors	100–200 Processors	>200 Processors
Compaq SC-667	1015	98–281	98–281	1334	125–188	100–174	88–127
Cray T3E-900	632	15–60	11–55	900	32–156	27–70	25–552
Cray T3E-1200	776	12–72	12–72	1200	36–68	30–89	40–657
HPTi ACL-667	1015	98–194	98–194	1334	195	176	—
IBM SP P2SC-120	338	17–93	10–93	480	57–95	—	—
IBM SP P2SC-135	440	19–91	12–78	540	16–48	82–115	—
IBM SP P2SC-160	447	23–108	14–92	640	6–200	70–205	—
IBM SP P3-HIGH-222	560	23–123	11–103	888	120	100	70
IBM SP P3-HIGH-375	1023	19–150	19–161	1500	142–500	115	105–111
IBM SP P3-THIN-200	576	21–111	10–93	800	78–250	45–400	20–360
IBM SP P3-THIN-375	1106	23–205	11–162	1500	141–500	75–170	40–113
SGI O2K-195	322	7–43	7–43	390	41–94	67	41–50
SGI O2K-250	412	13–76	13–76	500	26–200	10–112	30–85
SGI O2K-300	498	31–122	31–122	600	20–137	96–125	65–120
SGI O3K-400	683	50–208	50–208	800	114–218	104–111	85–108
SUN HPC10000-400	713	15–94	15–94	800	50–103	—	—

Note: The data for this table is a summary of the data from Tables 1 and 3.

5. References

1. McCalpin, J. "Equivalent MFLOPS" table for the STREAM Benchmark. Published electronically at http://www.cs.virginia.edu/stream.
2. Dongara, J. "Linpack Benchmark-Parallel" table for the Linpack Benchmark. Published electronically at http://www.netlib.org.
3. Dongara, J. "Linpack Benchmark" table for the Linpack Benchmark (N=1000). Published electronically at http://www.netlib.org.
4. "Complete NPB 2 Data 11/17/97: Graphs and Tables." Published electronically at http://www.nas.nasa.gov/Software/NPB.
5. "NAS Serial Benchmark Performance" table for the NAS Benchmarks (class A data set). Published electronically at http://www.nersc.gov/research/FTG/pcp/performance.html.
6. Levesque, J. M. Personal communication. IBM Research, 12 December 2000.
7. Hisley, D., C. Zoltani, and P. Satya-narayana. Personal communication. U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, and Raytheon, ARL-MSRC, 2000.
8. Saarinen, S. "Results of Some NAS Parallel Benchmarks on the NCSA Origin 2000–250 MHz (MPI)." Published electronically at http://archive.ncsa.uiuc.edu/Apps/Math/nas-irixse.html.
9. Zoltani, C. K., P. Satya-narayana, and D. Hisley. "Evaluating Performance of OpenMP and MPI on the SGI Origin 2000 With Benchmarks of Realistic Problem Sizes." ARL-TR-2324, U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, September 2000.
10. Cappello, F., and D. Etiemble. "MPI versus MPI+OpenMP on the IBM SP for the NAS Benchmarks." Published in the proceedings for SC2000 and electronically at http://www.sc2000.org, 2000.
11. Barrios, M., S. Andersson, G. Hanot, J. Hague, F. Johston, et. al. "Scientific Applications in RS/6000 SP Environments." Published electronically at http://www.ibm.com, 1999.
12. "SGI Origin Scaling for Density Functional Theory." Published electronically at http://archive.ncsa.uiuc.edu/SCD/Perf/Tuning/mp-scale, 2000.

13. "Achievements in Scalability." Published electronically at http://archive.ncsa.uiuc.edu/SCD/Perf/Tuning/mp_scale/riemann/nex.html, 1999.

14. "Scaling Comparisons of ZEUS-MP on Four Architectures." Published electronically at http://archive.ncsa.uiuc.edu/SCD/Perf/Tuning/mp_scale/zeus, 1999.

15. "Preconditioned Conjugate Gradient With Schwarz Richardson Preconditioner." Published electronically at http://archive.ncsa.uiuc.edu/SCD/straka/PerfAnalysis/Comps/pcg2.html.

16. Pfrommer, B. "Paratec Performance." Published electronically at http://mithril.ncsa.uiuc.edu./SCD/straka/PerfAnalysis/Apps/contrib.html.

17. Wallcraft, A. "Performance of the NRL Layered Ocean Model on Existing HPC Platforms." Published electronically at http://www.nrl.navy.mil/CCS/help/origin, 1997.

18. Gropp, W. D., D. K. Kaushik, D. E. Keyes, and B. F. Smith. "Performance Modeling and Tuning of an Unstructured Mesh CFD Application." Published in the proceedings for SC2000 and electronically at http://www.sc2000.org, 2000.

19. Brunner, R. K., J. C. Phillips, and L. V. Kalé. "Scalable Molecular Dynamics for Large Biomolecular Systems." Published in the proceedings for SC2000 and electronically at http://sc2000.org, 2000.

20. Drake, J. B., S. Hammond, R. James, and P. H. Worley. "Performance Tuning and Evaluation of a Parallel Community Climate Model." Published in the proceedings for SC99 and electronically at http://www.supercomp.org/sc99, 1999.

21. Carey, G. F., Dr. C. Harle, Dr. R. Mclay, and S. Swift. "MPP Solution of Rayleigh – Benard – Marangoni Flows." Published in the proceedings for SC97 and electronically at http://www.supercomp.org/sc97, 1997.

22. Wissink, A. M., and R. L. Meakin. "On Parallel Implementations of Dynamic Overset Grid Methods." Published in the proceedings for SC97 and electronically at http://www.supercomp.org/sc97, 1997.

23. Bashor, J. "NERSC Is Partner in Winning Supercomputing's Top Prize." Published electronically at http://www.nersc.gov, 1998.

24. Wallcraft, A. J. "Early Experience on NAVO's 2 TFLOPS Winterhawk II." Published in the electronic proceedings for SCICOMP 2000 at http://www.spscicomp.org/2000/presentations/Wallcraft, 2000.

25. Bettge, T., A. Craig, R. James, W. G. Strand, Jr., and V. Wayland. "Performance of the Parallel Climate Model on the SGI Origin 2000 and the Cray T3E." Published in the proceedings for the 41st Cray User Group Conference, Minneapolis, MN, 1999.

26. Pressel, D. M., J. Sahu, and K. R. Heavey. "Using Loop-Level Parallelism to Parallelize Vectorizable Programs." Published in the conference proceedings for High-Level Parallel Programming Models and Supportive Environments 6th International Workshop, HIPS 2001, San Francisco, CA, April 2001, Frank Mueller (Ed.), published by Springer as LNCS 2026.

27. Behr, M., D. M. Pressel, and W. B. Sturek Sr. "Comments on CFD Code Performing on Scalable Architectures." *Computer Models in Applied Mechanics and Engineering*, vol. 190, pp. 263-277, 2000.

28. Wallcraft, A. J. "Ocean Modeling on 1152 Cray T3E Processors." Published in the conference proceedings for the DOD High Performance Computing Modernizations Program 1999 Users Group Conference and electronically at http://www.hpmco.hpc.mil/Htdocs/UGS/UGC99/agenda.html.

29. Rood, E. P. "Complementary RANS and LES Computations for DDG-51 and Transition to DD-21 Acquisition." Published in the conference proceedings for the DOD High Performance Computing Modernization Program 1999 Users Group Conference and electronically at http://www.hpcmo.hpc.mil /Htdocs/UGS/UGC99/agenda.html.

30. Purtell, L. P. "Unsteady Hydrodynamics of the Maneuvering Submarine." Published in the conference proceedings for the DOD High Performance Computing Modernization Program 1999 Users Group Conference and electronically at http://www.hpmco.hpc.mil/Htdocs/UGS/UGC99/agenda .html.

31. "Parallel MM5 benchmarks, January-August 2000." Published electronically at http://box.mmm.ucar.edu/mm5/mpp/helpdesk/20000106.html.

32. Schraml, S. Personal communication. U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, 2000–2001.

33. Taft, J. R. "Achieving 60 **GFLOPS** on the Production CFD Code Overflow-MLP." Unpublished presentation from WOMPAT 2000 – Workshop on OpenMP Applications and Tools, San Diego, CA, 2000.

34. Roush, W. "256-processor SGI Origin 2000 System Announced at SC98." Published electronically in the January–February 1999 issue of *NASnews* at http://www.nar/nasa.gov/Pubs/NASnews/1999/01/o2ksidebar.html.

35. Moore, S. Personal communication. "Unpublished Results," University of Tennessee, Knoxville, TN, 2001.

36. Snavely, A. Personal communication. "Unpublished Results," San Diego Supercomputer Center, University of California, San Diego, CA, 2001.

37. Patel, J. "ParkBench and EuroBen Benchmarks on the AlphaServerSC." June 2000.

38. "SGI Origin 3000 Series Performance Report 1.0." Published electronically at http://www.sgi.com/developers/library/index.html. 20 September 2000.

39. Worley, P. "Evaluation of Early Systems." Published electronically at http://www.epm.ornl.gov/~worley/talks/ORNL-EES.SC2000, 2000.

40. Worley, P. "Kernel and Application Code Performance for a Spectral Atmospheric Global Circulation model on the Cray T3E and IBM SP." Presented at the NERSC Users' Group Meeting, Oak Ridge, TN, 6 June 2000 and published electronically at http://www.epm.ornl.gov/~worley/talks/ORNL-EES.SC2000, 2000.

41. Saito, T., A. Abe, and K. Takayama. "Benchmark of Parallelization Methods for Unstructured Shock Capturing Code." Proceedings of the 15th International Parallel and Distributed Processing Symposium (IPDPS 2001), San Francisco, CA, 23–27 April 2001.

42. Andersson, U., and G. Ledfelt. "A Billion Cells FD-TD Simulation of a Lightning Striking an Aircraft." Published electronically at http://www.nada.kth.se/~ledfelt/CEM/SC98/sc98.html, 1998.

43. Porter, D. H., P. R. Woodward, S. E. Anderson, K.-H. Winkler, and S. W. Hodson. "Numerical Simulation of Compressible Turbulence." Published electronically at http://www.lcse.umn.edu/research/lanlrun, 1997.

44. Shang, J. S., M. Wagner, Y. Pan, D. C. Blake, and C. J. Suchyta. "Strategies for Solving Time-Dependent Maxwell Equations on Multicomputers." Published electronically at http://www.hcpmo.hpc.mil/Htdocs/UGC/UGC98/papers/6a/index.htm.

45. Anderson, W. K., W. D. Gropp, D. K. Kaushik, D. E. Keyes, and B. F. Smith. "Achieving High Sustained Performance in an Unstructured Mesh CFD Application." Published electronically at http://www.llnl.gov/CASC/pubs/finalbell.pdf.

46. "SGI and ESI Power BMW for Faster Designs and Greater Safety." Published electronically at http://www.sgi.com/newsroom/press_releases/2000/october/bmw.html. 2000.

47. "T. J. Watson Research Center: Optimization Libraries." Published electronically at http://www.research.ibm.com/actc/Opt_Lib/Topic_OptLibraries.html. 2000.

48. Gupta, A., M. Joshi, and V. Kumar. "WSMP: A High-Performance Shared- and Distributed-Memory Parallel Sparse Linear Equation Solver." IBM Research Report RC22038 (98932), 20 April 2001.

49. "Proven Performance." Published electronically at http://209.238.152.199 /Clusterweb. 2000.

50. Briley, R., L. K. Taylor, and D. L. Whitfield. "Scalable Flow Simulations With Rotating Components." Spring 2001 NAVO MSRC Navigator, NAVAL Oceanographic Office Major Shared Resource Center, Stennis Space Center, MS, 2001.

51. Karniadakis, G. E. "Parallel Simulations of Flow-Structure Interactions." Published electronically at http://www.hpcmo.hpc.mil/Htdocs/Challenge /FY99/18.html.

52. Abate, J., P. Wang, and K. Sepehrnoori. "Parallel Compositional Reservoir Simulation on a Cluster of PCs." Published electronically at http://topeka.cpge.utexas.edu/papers/Cluster/Cluster.html.

53. Grimshaw, A. "Legion, Grids, and Clusters—the Future of High-Performance Computing." Presented at WGCC2000 in Tsukuba, Japan, published electronically at http://pdplab.trc.iwcp.or.jp.pdperf/wgcc-pdf/grimshaw.pdf.

54. Modi, A. "Unsteady Sparated Flow Simulations Using a Cluster of Workstations." Published electronically at http://bart.aero.psu.edu/thesis /thesis.pdf.

55. "FY00 Alliance Partnership Program Plan." Published electronically at http://archive.ncsa.uiuc.edu/people/jsheehan/brp/FY00.pdf.

INTENTIONALLY LEFT BLANK.

Glossary

ARL	U.S. Army Research Laboratory
CFD	Computational Fluid Dynamics
CTA	Computational Technology Area
DC	Distributed Center
DOD	Department of Defense
FFT	Fast Fourier Transform
GFLOPS	Billion Floating Point Operations per Second
HPC	High Performance Computing
HPCMP	High Performance Computing Modernization Program
MFLOPS	Million Floating Point Operations per Second
MIMD	Multiple Instruction Multiple Data
MPP	Massively Parallel Processor
MSRC	Major Shared Resource Center
NAS	Numerical Aerospace Simulation Systems Division of NASA Ames Research Center
NASA	National Aeronautics and Space Administration
NPB	NAS Parallel Benchmarks
SMP	Symmetric Multiprocessor

INTENTIONALLY LEFT BLANK.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
September 2002

3. REPORT TYPE AND DATES COVERED
Final, June 2000 – August 2001

4. TITLE AND SUBTITLE
Benchmarking the Benchmarks

5. FUNDING NUMBERS
665803.731

6. AUTHOR(S)
Daniel M. Pressel and Jelani Clay*

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
ATTN: AMSRL-CI-HC
Aberdeen Proving Ground, MD 21005-5067

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-2805

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)

10.SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
*Student Intern, Undergraduate Student at Prairie View A & M University, Prairie View, TX 77446-0397

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
Benchmarks can be useful in estimating the performance of a computer system when it is not possible or practical to test out the new system with an actual workload. In the field of high performance computing, some common benchmarks are the various versions of Linpack, the various versions of the Numerical Aerospace Simulation Systems Division of NASA Ames Research Center (NAS) benchmarks, and the STREAMS benchmark, as well as older and less frequently referenced benchmarks such as the Livermore Loops. There are also those who recommend estimating the performance based solely on the peak speed of the computer systems. Unfortunately, the per processor levels of performance measured using these benchmarks can vary by 1 to 2 orders of magnitude for the same system. Therefore, one has to ask, which benchmark(s) should we be looking at? This report attempts to answer that question by comparing the measured performance for a variety of real world codes to the measured performance of the standard benchmarks when run of systems of interest to the Department of Defense (DOD) High Performance Computing Modernization Program.

14. SUBJECT TERMS
benchmarking, high performance computing

15. NUMBER OF PAGES
33

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-102

INTENTIONALLY LEFT BLANK.